

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN**

Họ tên: Đàm Hồng Đức
MSSV: 19120481

BÁO CÁO ĐỒ ÁN KIẾN TRÚC PHẦN MỀM

Thành phố Hồ Chí Minh – Năm 2022

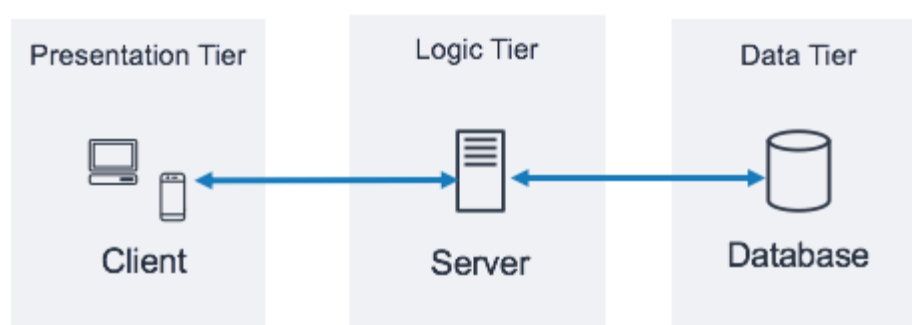
I. Thông tin nhóm

- Số lượng thành viên: 1
- Đồ án này xây dựng mới hoàn toàn cho môn: Kiến Trúc Phần Mềm
- Đồ án này sử dụng kiến trúc 3 layer:

MSSV	Họ Tên	Email	SDT
19120481	Đàm Hồng Đức	19120481@student.hcmus.edu.vn	0355211735

II. Kiến trúc 3-layer và 3-tier

1. Giới thiệu kiến trúc 3-tier



- 3-tiers là một kiến trúc kiểu client/server mà trong đó giao diện người dùng (UI-user interface), các quy tắc xử lý (BR-business rule hay BL-business logic), và việc lưu trữ dữ liệu được phát triển như những module độc lập, và hầu hết là được duy trì trên các nền tảng độc lập
- Đây là kiến trúc triển khai ứng dụng ở mức vật lý. Kiến trúc gồm 3 module chính và riêng biệt:
 - Presentation tier: hiển thị các thành phần giao diện để tương tác với người dùng như tiếp nhận thông tin, thông báo lỗi, ...
 - Business Logic tier: thực hiện các hành động nghiệp vụ của phần mềm như tính toán, đánh giá tính hợp lệ của thông tin, ... Tầng này còn di chuyển, xử lý thông tin giữa 2 tầng trên dưới.

- Data tier: nơi lưu trữ và trích xuất dữ liệu từ các hệ quản trị CSDL hay các file trong hệ thống. Cho phép tầng Business logic thực hiện các truy vấn dữ liệu
- Ưu điểm:
 - Phân loại rõ ràng các lớp có các nhiệm vụ khác nhau. Từ đó ta có thể quản lý và maintain project tốt hơn.
 - Dễ dàng phân loại các hành động tại Business.
 - Dễ dàng phân loại các hàm truy xuất tại Database, phân loại hàm theo table,...
 - Ứng dụng được cho các project lớn.

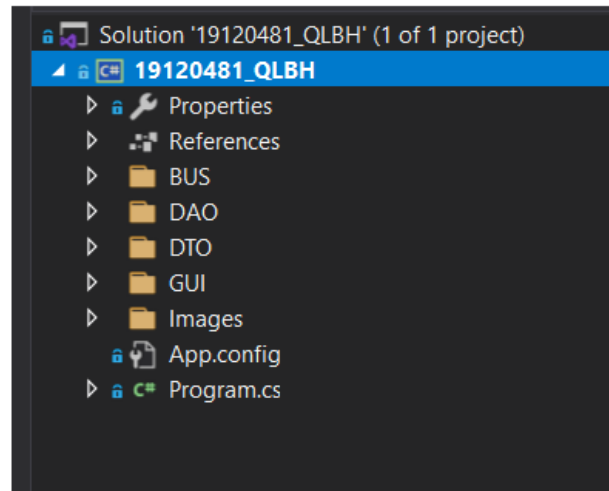
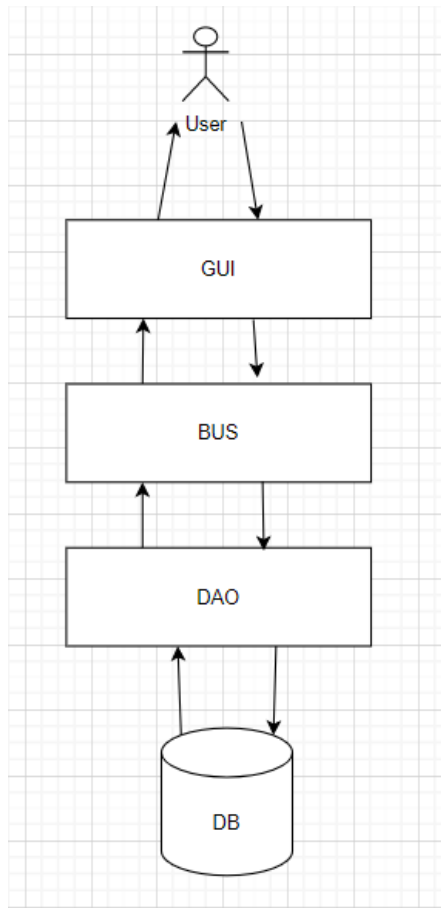
2. So sánh 3-layer và 3-tier

- Giống nhau: 3-layer giống 3-tier ở điểm là đều có vì cấu trúc phân chia giống nhau (presentation, bussiness logic, data).
- Khác nhau:
 - 3 tiers có tính vật lí, hệ thống chia ra thành các modules riêng biệt và lưu trữ ở những nơi khác nhau,...
 - 3 layer có tính logic, nghĩa là ta phân chia ứng dụng thành các phần (các lớp) theo chức năng hoặc vai trò một cách logic.
- Trong đồ án này, em sẽ áp dụng kiến trúc 3-layer

III. Tổng quan về đồ án

1. Kiến trúc

- Project này là 1 ứng dụng quản lý bán hàng dùng để quản lý việc bán hàng, sản phẩm, nhân viên,... cho cửa hàng.
- Project này sử dụng kiến trúc 3 layer:



- Project gồm có 4 thành phần chính:
 - GUI (presentation):
 - Lớp này có nhiệm vụ chính là giao tiếp với người dùng.
 - Gồm các thành phần giao diện winform, thực hiện các công việc như nhập liệu, hiển thị dữ liệu, kiểm tra tính đúng đắn dữ liệu trước khi gọi đến BUS layer.
 - BUS (business logic):
 - Là nơi đáp ứng các yêu cầu thao tác dữ liệu của GUI layer, xử lý chính nguồn dữ liệu từ GUI trước khi truyền xuống DAO layer.
 - Là nơi kiểm tra các ràng buộc, tính toàn vẹn và hợp lệ dữ liệu, thực hiện tính toán và xử lý các yêu cầu nghiệp vụ, trước khi trả kết quả về cho GUI layer.
 - DAO (data)
 - Lớp này có chức năng giao tiếp với hệ quản trị CSDL như thực hiện các công việc liên quan đến lưu trữ và truy vấn dữ liệu (insert, delete, update....).

- DTO
 - Lớp này được xây dựng thêm, mục đích là để chứa những dữ liệu được xây dựng dưới dạng lớp đối tượng.
- VD: Giải thích về việc phân chia các lớp (GUI, BUS, Dao, DTO) khi làm 1 tính năng cụ thể (ví dụ làm tính năng đăng nhập):
 - Ta tạo lần lượt tạo các file GUI_SignIn đặt trong GUI, file BUS_User đặt trong BUS, file DAO_User đặt trong DAO, file DTO_User đặt trong DTO.
 - GUI_SignIn là nơi chứa giao diện, người dùng sẽ tương tác với hệ thống như nhập tên đăng nhập, mật khẩu. Ngoài ra còn có 1 nút “Đăng Nhập” khi người dùng nhấn vào nút này hệ thống sẽ kiểm tra thông tin và gọi tới hàm Login trong lớp BUS

```
1 reference
private void btn_dangnhap_Click(object sender, EventArgs e)
{
    string tendn = textBox_tendangnhap.Text.Trim().ToString();
    string matkhou = textBox_matkhou.Text.Trim().ToString();

    // nếu chưa có dữ liệu
    if (tendn.Length == 0 | matkhou.Length == 0)
    {
        MessageBox.Show("Vui lòng nhập đầy đủ dữ liệu !!!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);
        return;
    }

    dtoUser = busUser.Login(tendn, matkhou);
}
```

- BUS_User có nhiệm vụ gọi tới lớp DAO và trả về kết quả lại cho GUI_SignIn

```
1 reference
public DTO_User Login(string username, string password)
{
    return daoUser.Login(username, password);
}
```

- DAO_User có nhiệm vụ giao tiếp với DB, thực hiện thao tác truy vấn kiểm tra thông tin đăng nhập và trả về kết quả

```
public DTO_User Login(string username, string password)
{
    string sql = "select * from M_USER where USERNAME = '" + username + "' and PASSWORD = '" + password + "'";
    DataTable tbl = DBAccess.GetDataToTable(sql);

    if (tbl.Rows.Count > 0)
    {
        int id = tbl.Rows[0].Field<int>(0);
        string fullname = tbl.Rows[0].Field<string>(3);
        DateTime dob = tbl.Rows[0].Field<DateTime>(4);
        int role = tbl.Rows[0].Field<int>(5);

        return new DTO_User(id, fullname, dob, role);
    }

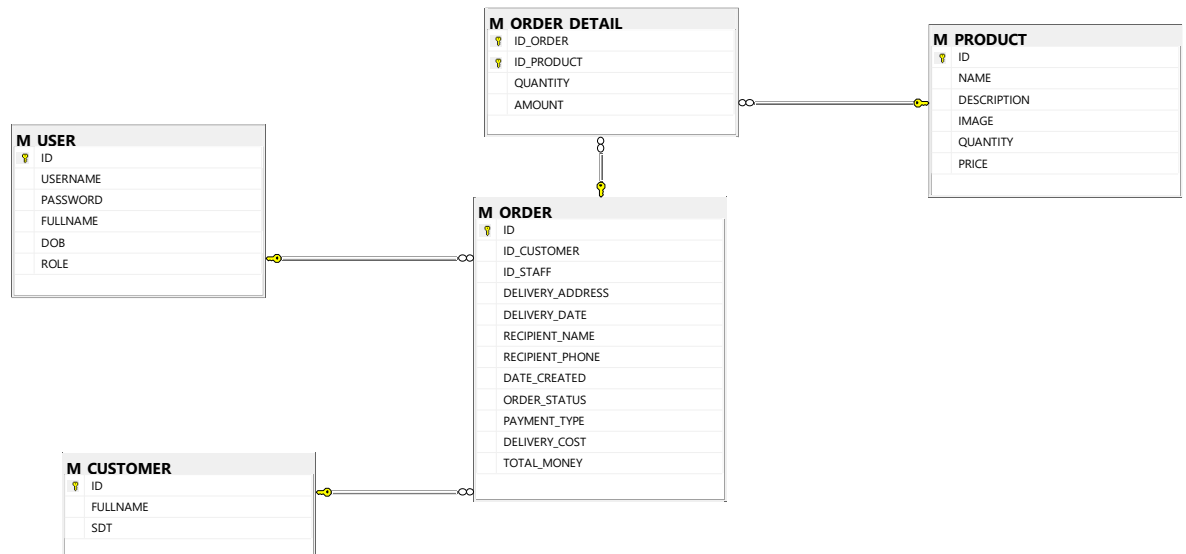
    return null;
}
```

2. Chức năng

- Project có 2 phân hệ là admin và staff, có các chức năng như sau:
 - Chức năng chung:
 - Đăng nhập
 - Đăng xuất
 - Phân hệ Admin:
 - Xem thông tin tất cả sản phẩm, tìm kiếm sản phẩm theo tên
 - Thêm, cập nhật, xóa 1 sản phẩm
 - Xem thông tin tất cả đơn hàng của cửa hàng
 - Xem thông tin tất cả Staff của cửa hàng
 - Tạo tài khoản, cập nhật thông tin 1 User
 - Phân hệ Staff:
 - Xem thông tin tất cả sản phẩm, tìm kiếm sản phẩm theo tên
 - Thêm 1 đơn hàng
 - Xem tất cả đơn hàng mà mình đã tạo

- Thêm 1 khách hàng mới khi tạo đơn hàng
 - Cụ thể: khi tạo đơn hàng sẽ có mục cho nhân viên chọn xem khách hàng đã có thông tin trong hệ thống chưa? nếu đã có thì NV tìm theo SĐT để ra thông tin khách hàng. Nếu chưa có thì NV nhập thông tin khách hàng -> khi tạo đơn hàng -> hệ thống đồng thời tạo mới luôn khách hàng

3. CSDL



- Hình trên mô tả CSDL của project gồm 5 bảng chính:
 - **M_USER**: lưu trữ thông tin user
 - Thuộc tính **ROLE** (phân hệ) có 2 giá trị:
 - 0 là admin
 - 1 là staff
 - **M_CUSTOMER**: lưu trữ thông tin khách hàng
 - **M_PRODUCT**: lưu trữ thông tin sản phẩm
 - **M_ORDER**: lưu trữ thông tin đơn hàng
 - Thuộc tính **ORDER_STATUS** (tình trạng đơn hàng) có 2 giá trị:
 - 0 là chưa thanh toán
 - 1 là đã thanh toán
 - Thuộc tính **PAYMENT_TYPE** (hình thức thanh toán) có 2 giá trị:
 - 0 là thanh toán bằng tiền mặt
 - 1 là chuyển khoản ngân hàng
 - **M_ORDER_DETAIL**: Lưu trữ thông tin chi tiết của 1 đơn hàng

IV. Demo, Github

- demo: <https://www.youtube.com/watch?v=t50e1A1ZNcY>
- github: <https://github.com/DAMHONGDUC/software-architecture>

V. Tham Khảo

- <https://viblo.asia/p/tim-hieu-ve-mo-hinh-3-lop-3-layer-Qbg5Qmyz5D8>
- <https://www.ibm.com/in-en/cloud/learn/three-tier-architecture>
- <https://www.ibm.com/topics/three-tier-architecture>